2018 年度 修士論文

組み換えを考慮した ゲノムグラフにおける アラインメントアルゴリズム

提出日：2019 年 2 月 1 日

指導教員：清水佳奈 教授
研究指導名：生命情報解析研究

早稲田大学 基幹理工学研究科
情報理工・情報通信専攻

学籍番号：5117F047-1

神保　元脩

目次

図目次

表目次

第 1 章

序論

1.1　研究背景

個人のゲノムを読み，個体差について調べることをリシークエンシング解析と呼ぶ．リシークエンシング解析では，シーケンサから読みだしたゲノム配列の断片（以降，リード配列）をヒトの代表配列（リファレンスゲノム配列）に照らし合わせて解析を行っていく．

ゲノム同士の対応付けを行う操作をアラインメントの計算と呼ぶ．リード配列とリファレンスゲノム配列のアラインメントの計算をうまく行うためには，リード配列がリファレンスゲノム配列と似ている必要がある．リード配列がリファレンスゲノム配列と似ていない場合，リード配列に対応するリファレンスゲノム配列上の位置を見つけることができない．

リファレンスゲノム配列はヒトの多様性を表現できているとは言い難い．そのため，多様性が認められる領域ではアラインメントを計算できないリード配列が増加してしまう問題が発生している．また，ヒトによってゲノム配列とリファレンスゲノム配列の類似度が異なるため，アラインメントの計算のしやすさが異なるという問題も発生している．これらの問題をリファレンスゲノム配列によるバイアスと呼んでいる [4]．リファレンスゲノム配列によるバイアスは解析フロー全体に影響を及ぼしてしまう可能性があり，大きな問題である．

この問題の解決策として，ゲノムグラフが考案されている．ゲノムグラフとは，ゲノム配列にグラフ構造を導入したものである．ゲノムグラフで表現されたリファレンスゲノム (リファレンスゲノムグラフ) は，ゲノムの多様性やあいまいさを表現することができるため，リファレンスゲノムによるバイアスを減らすことが期待されている．

1.2　研究目的

リファレンスゲノム配列におけるリード配列のアラインメント計算は，最もリード配列と似ている部分文字列を探す操作である．一方，リファレンスゲノムグラフにおけるリード配

列のアラインメント計算は，最もリード配列と適合するゲノムグラフ上のパスを探す操作となる．

ゲノムグラフにおけるアラインメントの計算手法はいくつか提案されている [5][7]．これらの手法は，従来のリファレンスゲノム配列におけるアラインメントの計算手法を拡張したものであり，塩基単位のみでの確率モデルを考えている．しかし，ゲノムグラフにおけるアラインメントの計算とは，リード配列と対応するゲノムグラフ上のパスを探す操作である．そのため，事前に観測されているパスについても考慮してアラインメントの計算を行うべきであると考えられる．ゲノムグラフを扱うことのできるツールの 1 つである vg[1] ではパスの情報をアラインメントの計算に利用している．しかし，パスの情報の活用は従来のアラインメントの計算手法で算出したスコアに補正を加える程度であり，アラインメントの結果は従来のアラインメント計算手法に大きく依存している．

本研究では，リファレンスゲノムグラフ上のパスの情報を活用しながらアライメントの計算を行う新しいアルゴリズムを提案する．

1.3 本研究の貢献

以下に，本論文で達成したことを列挙する．

- ゲノムグラフに対するアラインメントの確率モデルの提案
- 従来手法と提案手法の確率モデルとアルゴリズムの比較
- 塩基単位のコストと組み換えのコストの和を最小化するアラインメントの計算

1.4 本論文の構成

以下に本論文の構成を示す．第 1 章では，研究背景，研究目的と本研究の貢献について述べる．第 2 章では，本論文における用語の定義とゲノムグラフに対するアラインメント計算の関連研究について述べる．第 3 章では，提案手法のアルゴリズムの詳細について述べる．第 4 章では，提案手法を評価するために行った実験の詳細と結果について述べ，関連研究との比較や提案手法の問題点について検討を行う．第 5 章では，本研究の結論について述べる．

第 2 章

関連研究

2.1　用語の定義

関連研究の説明を行う前に，本論文で用いる用語の定義を述べる．従来研究と定義が異なる場合もあるが，本論文内ではこの節での定義を用いて説明を行う．

ゲノムグラフは頂点と辺の集合である．頂点は固有の番号（頂点 ID）と 1 塩基を持つ．頂点 ID が i の頂点を頂点 i と呼ぶ．辺は 2 つの頂点間を結ぶ有向辺である．パスはゲノムグラフ上の経路を表すもので，頂点 ID の列として表現される．ただし，パスは辺の存在する頂点間のみを通過する．ハプロタイプは塩基列であり，‘A’,‘T’,‘G’,‘C’ の 4 種類の文字からなる文字列で表現される．アラインメントはハプロタイプとパスの組である．アラインメントに含まれるハプロタイプとパスで表現される塩基列が等しいとは限らない．ハプロタイプとパスで表現される塩基列の差は置換・挿入・欠損が加わったと解釈をする．アラインメントのスコアとは，アラインメントがゲノムグラフとどの程度合致しているかを示す指標であり，置換・挿入・欠損といった塩基単位でのコストとパスに対するコストから求められる．アラインメントを計算するとは，あるハプロタイプに対して最もスコアが良くなるアラインメントを探索することである．

2.2　ゲノムグラフにおけるアラインメントアルゴリズム

ゲノムグラフに応用できるアラインメントアルゴリズムとして，従来のペアワイズアラインメントの計算手法をグラフ構造に対応できるように拡張したものが提案されている．ここでは動的計画法（以降 DP）を用いる手法とダイクストラ法を用いる手法の 2 種類の手法について説明をする．

2.2.1 動的計画法を利用する手法

Partial Order Alignment(以降 POA)[5] では，文字列同士のペアワイズアラインメントを計算するための漸化式を，グラフの構造に従って変形している．計算式は以下の通りである．

$$\begin{aligned} &\mathsf{DP}[0][0] = 0 \\ &\mathsf{DP}[i][j] = \min \begin{cases} \mathsf{DP}[i][p] + D \\ \mathsf{DP}[i-1][j] + I \\ \mathsf{DP}[i-1][p] + \mathsf{Score}(q[i-1], G[j]) \end{cases} \end{aligned} \tag{2.1}$$

$\mathsf{DP}[i][j]$ はクエリの i 文字目とゲノムグラフの頂点 j までのアラインメントのスコアを表している．頂点 p は頂点 j の直前にあるすべての頂点を表している．I は挿入のコストを，D は欠損のコストを表している．$q[i]$ はクエリハプロタイプの i 番目の塩基を，$G[j]$ はゲノムグラフの頂点 j に対応する塩基を表している．$\mathsf{Score}(a, b)$ は塩基 a と b の置換のコストを返す関数を表している．

POA の計算手法は文字列同士のペアワイズアラインメントとほぼ変わらず，計算量がグラフ全体の文字数とクエリの文字数にそれぞれ比例する程度と軽い．そのため，大きなゲノムグラフに対してもアラインメントを計算することができる．さらに，スコアの上限値をあらかじめ決めておくことで，DP 行列の中で計算する必要のあるセルを減らすことができ，高速化することができる [12]．しかし，POA では DAG に対するアラインメントのみを対象としており，ループを含んだグラフに対するアラインメントの計算を行うことはできない..

2.2.2 ダイクストラ法を利用する手法

ループを含んだグラフに対するアラインメント手法は Rautiainen らによって提案されている [7]. Rautiainen らの手法では，POA をループを含んだグラフに対応できるように拡張を行っている．DAG に対するアラインメントの計算では各セルの値が前のセルの値のみに依存していることが明らかなので DP で計算を行うことが可能である．一方でループを含んだグラフに対するアラインメントの計算では，セルの値がそれより先のセルの値に依存する可能性があるので，DP では最適なスコアを計算することができない．Rautiainen らの手法では，ダイクストラ法によって最適なスコアを計算すると同時に，最適なスコアをとる経路に対応する塩基列とパスをアラインメント結果として出力している．ダイクストラ法を利用しているのでコストの値に負の値が含まれないことが制約となる．

2.3 ゲノムグラフにおけるパスの確率モデル

複数のハプロタイプが事前に観測されている時，それらのハプロタイプから新しいハプロタイプが組み換えによって生成される確率のモデルが Li らにより考案されている [6]. Li らの確率モデルでは，事前に観測されたハプロタイプ群はマルチプルアラインメントである．新しいハプロタイプの各塩基が事前に観測されているハプロタイプのいずれかから受け継いだものであると仮定し，近隣の塩基は同じ親から受け継いでいる可能性が高いということを想定した確率モデルとなっている．

このモデルをゲノムグラフに拡張し，効率よく計算する手法が Rosen らによって考案されている [8]. Rosen らの手法は，Li らの手法と比べてハプロタイプをアラインメントに置き換えている点と，置換・挿入・欠損を確率モデルで考慮していない点が異なっている．つまり，アラインメントのハプロタイプはパスが表す塩基列と完全に一致するという条件を置いたうえで，新しいパスが既存のパスをどのように組み換えたら発生するかという問題に置き換えている．Rosen らの手法は，部分パスを効率よく列挙できる手法である GBWT[11] を活用することで，最悪の場合でもハプロタイプの長さと事前に観測されているアラインメントの本数に線形比例する時間で計算できる．

2.4 vg でのアラインメント

ゲノムグラフ用のツールである vg にはアラインメント計算用のコマンドとして vg map と vg mpmap の 2 つが用意されている．ここでは，その 2 つのコマンドに実装されているアルゴリズムの概要について述べる．

2.4.1 vg map

vg の map コマンドでは大きく分けて，seed の探索・seed のクラスタリング・サブグラフに対するアラインメント計算・最適なアラインメント結果の選出の 4 つの処理が行われる．以下でそれぞれの処理の説明を記載する．

seed の探索

vg において seed は Maximum Exact match(以降 MEM) と呼ばれる．MEM はクエリとゲノムグラフの共通部分文字列であり，それ以上延長できないという性質を持つ．ゲノムグラフに含まれる部分文字列を効率よく検索できるデータ構造である GCSA2[10] と，pBWT[2] のアルゴリズムの一部である，Set Maximal Match を計算するアルゴリズムを組み合わせることで MEM を計算している．

seed のクラスタリング

MEM をゲノムグラフ上で近い位置にあるもの同士で分類し，複数の MEM クラスタを生成する．次に，MEM クラスタに含まれるすべての MEM を含むサブグラフを各 MEM クラスタに対して生成する．MEM の位置関係を推定するためには，事前に観測されたアラインメントが利用されている．アラインメント上に 2 つの MEM m_1 と m_2 が存在する時，m_1 と m_2 の間に存在する塩基数を m_1 と m_2 の距離としている．

サブグラフに対するアラインメント計算

各サブグラフに対して，DAG 化を行う．事前に，ループを回る上限回数を決めておき，その上限回数までのループに関しては元のゲノムグラフと整合性が取れるような DAG を生成する．ただし，reference fasta ファイルと vcf ファイルからゲノムグラフを生成する手法（vg でゲノムグラフを作る際，一般的に用いられる手法）では DAG が生成されることが保証されているので，多くの場合 DAG 化の処理は必要無い．DAG 化した各サブグラフに対して POA を実行する．アラインメント結果の候補数はサブグラフの数と等しくなる．

最適なアラインメント結果の選出

POA によって得られたアラインメント計算のスコアに補正を加える．具体的には，得られたアラインメントスコアに，Rosen らのアルゴリズムで計算したパスの生成確率を元にしたコストを加算する．補正をかけたアライメントスコアが最も高いアラインメント結果が，vg map での出力となる．

2.4.2 vg mpmap

mpmap は multipath mapper の略である．seed のクラスタリングまで，vg mpmap は vg map とほぼ同じ処理を行う．vg map では，各サブグラフに対してアラインメント結果の候補が 1 つだけであるのに対して，vg mpmap では各サブグラフに対して複数のアラインメント結果の候補を持つことができる設計となっている．そのため，vg map では見つけることのできなかったアラインメント候補を見つけられる可能性がある．しかし，vg mpmap では MEM からアラインメントを伸長するアルゴリズムを用いているため，アラインメント結果が MEM に大きく依存する設計となっている．例えば，リードにシーケンスエラーがあり MEM がゲノムグラフ上で間違った分岐上に当たってしまった場合は，正しいアラインメントを計算することができなくなる．

第 3 章

提案手法

3.1　導入

本章では，事前に観測されているアラインメントの情報を活用してゲノムグラフ上でアラインメントの計算を行うアルゴリズムの提案を行う．従来，アラインメントの計算で用いられるコストは HMM の遷移確率と出力確率に対応しており，アラインメントは HMM 上の最尤パスに相当する．本章では，まず本研究の目指していることを紹介する．次に，提案手法におけるアラインメントの計算の背景にある確率モデルを紹介する．そのあとアルゴリズムの詳細について説明を行っていく．

なお，本研究では POA と同様，DAG に対するアラインメントの計算のみを考えるものとし，ループを含んだグラフは扱わないものとする．ただし，本手法を拡張することでループを含んだグラフに対してもアラインメントの計算を行うことができるようになる．拡張についてはこの章の最後で述べる．

3.2　理想的なアラインメント

本研究での理想は，塩基の置換・欠損・挿入によるコストとパスのコストの和を最小にするアラインメントを発見することである．塩基の置換・欠損・挿入によるコストを最小にするアラインメントは POA によって，クエリの長さとグラフの文字数に線形比例する時間で計算を行うことができる．また，パス P のコストは事前に観測されているアラインメントの組み換えによってパス P が得られる確率を元に計算される．この確率は Rosen らのアルゴリズムで計算できる．つまり，塩基単位でのコストと組み換えによるコストを別々に計算することは，それぞれ線形時間で終えることが可能である．

しかし，塩基単位のコストが最小となるアラインメントと塩基単位のコストとパスのコストの和が最小となるアラインメントが同じであるという保証はない．つまり，塩基単位のコ

ストを最小にするアラインメントを効率よく見つけることができ，さらにそのアラインメントのパスのコストを効率よく計算することができたとしても，2 種類のコストの和を最小にするアラインメントを効率よく見つけることができるわけではない．vg でアラインメントの計算を行う際は，まず塩基単位のコストのみを考えてアラインメントの候補を複数列挙している．しかし，その列挙した中に最良のアラインメントが含まれているとは限らない．

2 種類のコストの和が最小となるアラインメントを効率よく計算する手法は現在確立されていない．例えば，POA を単純に拡張して塩基単位でのコストと組み換えによるパスのコストを元にアラインメントを以下のように計算しようとしたとする．

$$
\begin{aligned}
&\mathsf{DP}[0][0] = 0 \\
&\mathsf{DP}[i][j] \\
&= \min \begin{cases} \mathsf{DP}[i][p] + D + \mathsf{PathCost}(\mathsf{DP}[i][p].path + j) \\ \mathsf{DP}[i-1][j] + I + \mathsf{PathCost}(\mathsf{DP}[i-1][j].path) \\ \mathsf{DP}[i-1][p] + \mathsf{Score}(q[i-1], G[j]) + \mathsf{PathCost}(\mathsf{DP}[i-1][p].path + j) \end{cases}
\end{aligned} \tag{3.1}
$$

PathCost はパスを引数にとり，パスのコストを返す関数を表している．DP 行列の各セルの $path$ フィールドにそれまで通過してきたパスを記録しているものとする．パス P に頂点 n を加算することで，パス P を頂点 n で延長したものとみなすことにしている．

式 3.1 ではそれなりに良いアラインメントを得られる可能性はあるが最適なアラインメント結果を得られる保証はない．なぜならば，DP で最適パスを見つけるためには，最適パスの部分パスは必ずその区間の最適パスであるという条件を満たしている必要があるが，2 種類のコストの和を最小化する問題ではそのような条件を満たすことができないからである．

3.3 提案手法の確率モデル

前節の理想的なアラインメントの計算では，事前に観測されたアラインメントの組み換えの全パターンを考慮したパスの生成確率を，パスのコストであると考えていた．しかし，理想的なアラインメントの計算時間は，ゲノムグラフの分岐数の増加に対して指数関数的に増加すると想定され実用的ではない．

ここで，本手法ではパスのコストの代替となるコストを，組み換えのパターンの中で最も組み換えの回数が少ないパターンの生成確率から求めることにする．この代替コストを組み換えコストと呼ぶことにする．組み換え確率が低い状況を想定するならば，パスのコストと組み換えコストはおおよそ一致すると考えられる．また，パスのコストを組み換えコストに置き換えることで，考える必要のある組み換えのパターン数を非常に少なくすることができる．そのため，効率よく計算することができる．

本研究では，ProfileHMM[3] をゲノムグラフに拡張した確率モデルを提案し，この確率モ

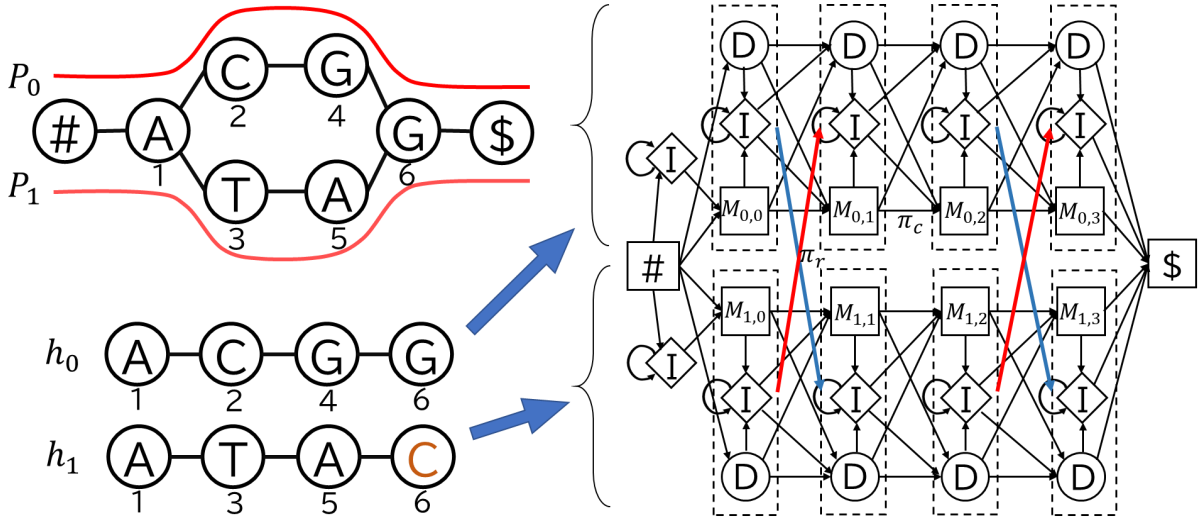


図 3.1　左上：ゲノムグラフと，ゲノムグラフ上でとりうる 2 種類のパス P_0 と P_1 を示している．各頂点の丸の中に書いてある文字はその頂点に対応する塩基であり，丸の下に書いてある数字は頂点 ID である．#が書かれている頂点は開始ノードで$が書かれている頂点は終端ノードである．左下：2 つのアラインメント h_0 と h_1 を示している．例えばアラインメント h_1 のハプロタイプは “ATAC” である．h_1 のパスは “1,3,5,6” であるので，このハプロタイプは P_1 に置換が 1 回起きた結果であると解釈することができる．右：左上のグラフと左下のアラインメントが与えられた場合の提案手法における確率モデルを示している．上半分が h_0 から作られた ProfileHMM，下半分が h_1 から作られた ProfileHMM である．M が書かれている正方形は，h_* との一致・不一致の状態を示す．I は挿入，D は欠損を表す状態である．青色の矢印は h_0 から h_1 へ向かう組み換えを表しており，赤色の矢印は h_1 から h_0 へ向かう組み換えを表している．組み換えの確率は π_r，組み変わらない確率は π_c と表している．この確率は頂点によって異なっており，例えば頂点 ID が 2 の頂点から頂点 ID が 4 の頂点へ向かう辺はアラインメント h_0 しか通過していないため，組み換えが起きる確率 $\pi_r = 0$ となる．

デル上での最尤パスをアラインメントとして出力するアルゴリズムを提案する．事前に観測されている各アラインメントに 1 つの ProfileHMM を構築し，ゲノムグラフの構造に従って ProfileHMM 間を遷移できるモデルを考案した．組み換えを表現できる ProfileHMM として jpHMM が提案されている [9]．本手法でも，jpHMM と同様に ProfileHMM 間に組み換えを表す遷移確率を与えている．ゲノムグラフの形状と事前に観測されているパスの本数によって遷移確率を変化させている点が本手法と jpHMM の違いである．図 3.1 に提案するモデルの概要を示す．提案手法の確率モデルをたどることで，置換・挿入・欠損・組み換えを考慮したアラインメントの生成確率を計算することができる．

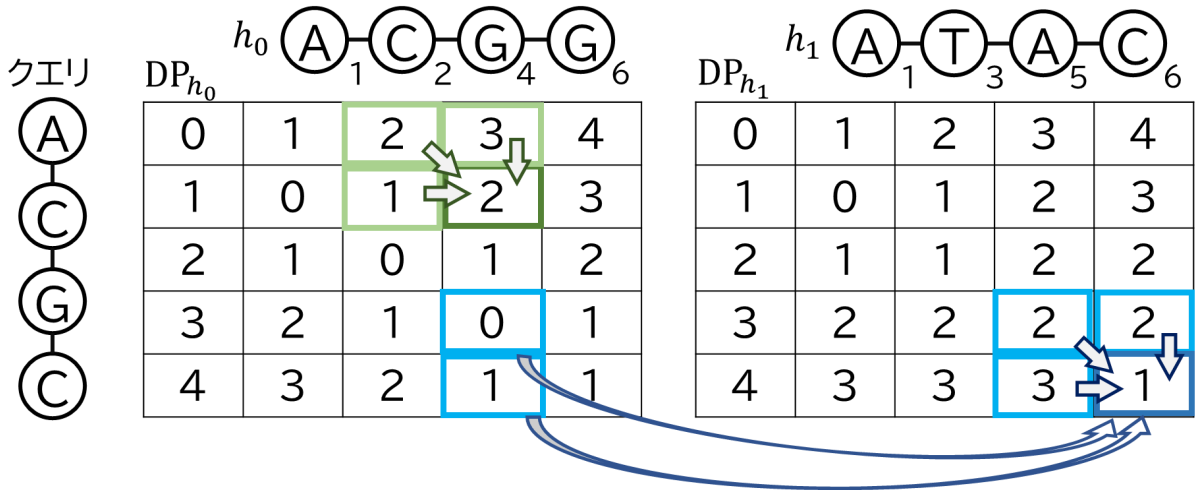


図 3.2　図 3.1 に示したゲノムグラフに対して，提案手法でクエリハプロタイプ”ACGC”のアラインメントを計算する様子を表している．左側がアラインメント h_0 に対応する DP 行列 DP_{h_0}，右側が h_1 に対応する DP 行列 DP_{h_1} となっている．縦方向の矢印は挿入を，横方向の矢印は欠損を，斜め方向の矢印は一致・不一致を，別の DP 行列へ向かう矢印は組み換えを表している．置換・挿入・欠損・組み換えのコストはすべて 1 としている．緑のセルがある列は頂点 ID が 4 の頂点に対応している．頂点 ID が 4 の頂点は頂点 ID が 2 の頂点からのみ辺が入射していて，その辺を通るアラインメントは h_0 のみなので，緑のセルを計算する場合は組み換えを考える必要はない．一方で，青のセルがある列は頂点 ID が 6 の頂点に対応しており，頂点 ID が 4 の頂点と 5 の頂点から辺が入射している．これらの辺を h_0，h_1 共に通過するため，青のセルを計算する場合は組み換えを考慮する．DP_{h_0} の右下のセルの 1 は h_0 に 1 文字置換が起こったアラインメントのスコアを表している．DP_{h_1} の右下のセルの 1 は 3 文字目まで h_0 と一致し 4 文字目は組み換えの後 h_1 と一致しているアラインメントのスコアを表している．

3.4　提案手法のアルゴリズム

アフィンギャップスコアのように挿入・欠損が開始する時のコストと連続する時のコストを変えるために，1 つのアラインメント h_* あたり 3 つの DP 行列 $\mathrm{DP}^M_{h_*}, \mathrm{DP}^D_{h_*}, \mathrm{DP}^I_{h_*}$ を計算する必要がある．$\mathrm{DP}^D_{h_*}$ が欠損の状態を，$\mathrm{DP}^I_{h_*}$ が挿入の状態を表す DP 行列である．クエリの長さを n，h_* のハプロタイプの長さを m とすると，$\mathrm{DP}^*_{h_*}$ の行数は $n+1$，列数は $m+1$ となる．アラインメント h_* のハプロタイプの i 文字目を $h_*[i]$ で表し，対応する頂点 ID を $N(h_*[i])$ で表す．また，DP 行列 $\mathrm{DP}^*_{h_*}$ の i 列目 j 行目のセルを $\mathrm{DP}^*_{h_*}[i][j]$ と表す．

アラインメントを計算するための漸化式は以下の通りである．

$$
\begin{aligned}
&\mathsf{DP}^{*}_{h_*}[0][0] = 0 \\
&\mathsf{DP}^{M}_{h_*}[i][j] = \min \left\{ \begin{array}{l}
\mathsf{DP}^{M}_{h_*}[i-1][j-1] + \mathsf{Score}(q[i-1], h_*[j-1]) + C_{N(h_*[j-1])} \\
\mathsf{DP}^{I}_{h_*}[i-1][j-1] + \mathsf{Score}(q[i-1], h_*[j-1]) + C_{N(h_*[j-1])} \\
\mathsf{DP}^{D}_{h_*}[i-1][j-1] + \mathsf{Score}(q[i-1], h_*[j-1]) + C_{N(h_*[j-1])} \\
\mathsf{DP}^{M}_{h_p}[i-1][k] + \mathsf{Score}(q[i-1], h_*[j-1]) + R_{N(h_*[j-1])} \\
\mathsf{DP}^{I}_{h_p}[i-1][k] + \mathsf{Score}(q[i-1], h_*[j-1]) + R_{N(h_*[j-1])} \\
\mathsf{DP}^{D}_{h_p}[i-1][k] + \mathsf{Score}(q[i-1], h_*[j-1]) + R_{N(h_*[j-1])}
\end{array} \right. \\
&\mathsf{DP}^{I}_{h_*}[i][j] = \min \left\{ \begin{array}{l}
\mathsf{DP}^{M}_{h_*}[i-1][j] + I^{d}_{h_*[j-1]} \\
\mathsf{DP}^{I}_{h_*}[i-1][j] + I^{e}_{h_*[j-1]}
\end{array} \right. \\
&\mathsf{DP}^{D}_{h_*}[i][j] = \min \left\{ \begin{array}{l}
\mathsf{DP}^{M}_{h_*}[i][j-1] + D^{d}_{h_*[j]} + C_{N(h_*[j-1])} \\
\mathsf{DP}^{D}_{h_*}[i][j-1] + D^{e}_{h_*[j]} + C_{N(h_*[j-1])} \\
\mathsf{DP}^{M}_{h_p}[i][k] + D^{d}_{h_*[j]} + R_{N(h_*[j-1])} \\
\mathsf{DP}^{D}_{h_p}[i][k] + D^{e}_{h_*[j]} + R_{N(h_*[j-1])}
\end{array} \right.
\end{aligned}
\tag{3.2}
$$

頂点 $N\left(h_p[k]\right)$ を始点とし頂点 $N\left(h_*[j]\right)$ を終点とするような辺が存在するすべての $h_p[k]$ について計算を行う．I^d は挿入開始のコスト，I^e は挿入連続のコスト，D^d は欠損開始のコスト，D^e は欠損連続のコスト，C は組み換えをしないコスト，R は組み換えを行うコストを表している．コストの添え字に頂点の ID が書かれており，各頂点によって値が異なることを表している．なお，各コストの値は負の値ではないとする．

上の式はコストパラメータが非常に多く複雑であるため厳密に計算を行うことが困難である．そのため，本論文では，置換・挿入・欠損・組み換えに関するコストをそれぞれ一定の値であると仮定して話を進める．置換・挿入・欠損・組み換えのコストがそれぞれ一定の値であれば DP 行列は事前に観測された各アラインメントに対して 1 つでよいため，以下の式で表すことができる．

$$
\begin{aligned}
&\mathsf{DP}_{h_*}[0][0] = 0 \\
&\mathsf{DP}_{h_*}[i][j] = \min \left\{ \begin{array}{l}
\mathsf{DP}_{h_*}[i-1][j] + I \\
\mathsf{DP}_{h_*}[i][j-1] + D + C \\
\mathsf{DP}_{h_*}[i-1][j-1] + \mathsf{Score}(q[i-1], h_*[j-1]) + C \\
\mathsf{DP}_{h_p}[i][k] + D + R \\
\mathsf{DP}_{h_p}[i-1][k] + \mathsf{Score}(q[i-1], h_*[j-1]) + R
\end{array} \right.
\end{aligned}
\tag{3.3}
$$

提案手法のアラインメントの計算の流れを図 3.2 に示した．最適なアラインメントのスコアは，ゲノムグラフの終端ノードと隣接する頂点に対応する DP 行列の列の最下段の値の中で最小の値となる．アラインメントそのものを得るためには最小の値をとるセルからトレースバックを行う必要がある．トレースバックを行うためには最小値の計算を行う際に，各セルごとに最小値をとったセルがどこかを記録しておく必要がある．トレースバックで開始

ノードまでたどることで，クエリハプロタイプが事前に観測されたどのアラインメントが組み換えられたものか，どのような編集が行われたかについて解釈することができる．

3.5　提案手法の高速化

式 3.3 をそのまま計算すると，例えば分岐のない直線のゲノムグラフに対してアラインメントを計算する際，すべての頂点についてすべてのアラインメントとの組み換えを考慮して漸化式を計算することになる．しかし，本手法においては最も組み換えの回数が少ないパターンのみを考えているので，これは無駄である．直線のゲノムグラフのとりうるパスは 1 通りしかないので，そのパスを持つアラインメントが事前に観測されているのであれば，組み換えを考える必要はないからである．

ここでは式 3.3 を効率よく計算するための 2 つの高速化手法について説明を行う．

3.5.1　アラインメントの統合

ここまでは，事前に観測された各アラインメントに対して ProfileHMM を構築してきた．しかし，ここでは事前に観測された複数アラインメントに対し 1 つの ProfileHMM を構築する手法を提案する．

ゲノムグラフにおけるアラインメントはパスと塩基列の組であると定義した．ここで，同じパスを持つアラインメントらは，1 本の配列上で整列された塩基列であるとみなせるので，マルチプルアラインメントと等価である．ProfileHMM はマルチプルアラインメントを表現するために考案された確率モデルであり，同じパスを持つアラインメントを 1 つの ProfileHMM にまとめることは自然である．

アラインメントを統合することにより，ProfileHMM の個数は事前に観測されているアラインメントの個数以下に抑えることができる．事前に観測されているアラインメントの個数は多いものの，アラインメントが持つパスの種類は少ないといった状況においては大幅な高速化が期待できる．

3.5.2　最小限の組み換え

式 3.3 では，複数の事前に観測されたアラインメントが通過する頂点すべてに対しては組み換えを考慮して計算を行っている．しかし，ゲノムグラフの形状とコストパラメータによっては，そのような頂点すべてで組み換えを考える必要はない．

事前に観測されているアラインメントの本数が増減せず，各アラインメントに対応する ProfileHMM に対して，置換・挿入・欠損のコストが等しい区間においては組み換えを考える必要はない．これは，上記の条件を満たす区間内で組み換えが起きているパターンが最適

なスコアをとることはないからである．

各コストを ProfileHMM 毎に設定できるようにすることで自由度の高いアラインメントの計算を行うことができる．一方で，各コストを ProfileHMM によらず一定にすることで計算時間を短縮することが可能となる．

3.6 計算量

提案手法では DP 行列のすべてのセルに対して式 3.3 を計算することになるので，空間計算量は DP 行列のセルの個数と等しくなる．クエリハプロタイプの長さを N，すでに観測されているアラインメントの集合を H，アラインメント h のハプロタイプの長さを $|h|$ とすると，空間計算量は $O\left(N\sum_{h\in H}|h|\right)$ となる．

時間計算量はゲノムグラフの形状によって異なる．これは，ゲノムグラフの分岐によって，1 つのセル当たりの計算時間が異なるからである．ゲノムグラフが直線上かつ，事前に観測されているアラインメントが途中から始まったり終わったりせず，置換・挿入・欠損がない場合，計算量は $O(NM)$ と最も小さくなる．M は事前に観測されているアラインメントのハプロタイプの長さである．

最も時間計算量が多くなる場合は，各頂点に対して分岐が存在し事前に観測されたアラインメントのパスに重複が存在しない場合であり，$O\left(N|H|\sum_{h\in H}|h|\right)$ となる．

3.7 提案手法の拡張

ここまではゲノムグラフが DAG であることを想定してアルゴリズムを説明してきた．しかし，本節ではアルゴリズムを拡張することでループを含んだゲノムグラフに対してもアラインメントの計算が可能であることを説明する．

3.7.1 拡張確率モデル

拡張された確率モデルを，図 3.3 に示す．基本的には図 3.1 で示した確率モデルと同じだが，アラインメントがループを通過する場合はループを通過する前と通過した後が組み換えで接続される点が異なっている．ループ部を通過するアラインメントが観測されている場合は，アラインメントの後方から前方へ向かう遷移が存在する．組み換えのコストの値によっては，置換・挿入・欠損が組み換えとして解釈される可能性がある．

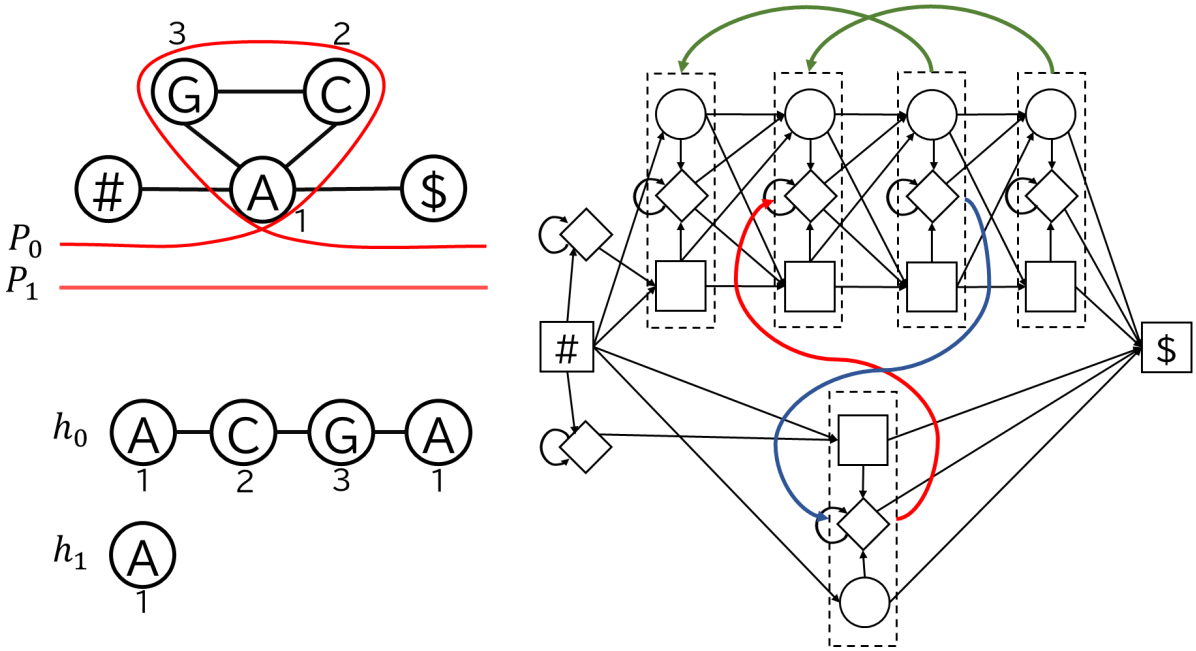


図 3.3　左上：ループを含むゲノムグラフと，ループ部を通過するパス P_0 と通過しないパス P_1 を示している．左下：事前に観測されている 2 つのアラインメントを示している．右：左のゲノムグラフとアラインメントから生成された拡張確率モデルを示している．各図形の表す状態は図 3.1 と同じである．頂点 ID が 1 の頂点を h_0 は 2 回，h_1 は 1 回通過しており，その頂点での組み換えを表しているのが黒以外の色のついた矢印である．青の矢印は h_0 から h_1 への組み換え，赤の矢印は h_1 から h_0 への組み換え，緑の矢印は h_0 内での組み換えとなっている．

3.7.2　拡張アラインメントアルゴリズム

拡張された確率モデルでは後方から前方への遷移が存在するため，動的計画法でアラインメントを計算することはできない．しかし，Rautiainen らの手法と同じようにダイクストラ法を用いることでアラインメントを計算することが可能である．

第 4 章

実験

4.1 実験概要

提案手法の性能を評価するために，提案手法を実装し実験を行った．計算には Intel(R) Xeon(R) CPU E5-2643 v3 @ 3.40GHz，メモリ容量 128GB のマシンを用いた．計算はシングルスレッドで行った．

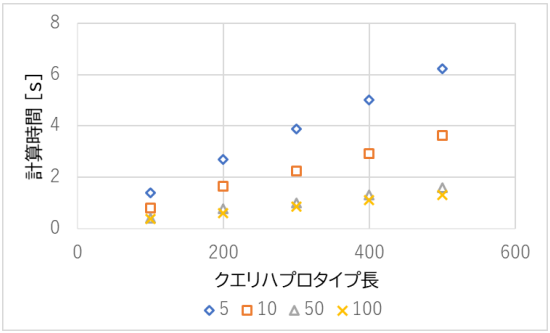
提案手法がリソースをどの程度消費するかを計測する実験と，提案手法が既存手法と比べてアラインメント精度がどの程度異なるかを計測する実験を行った．計測結果は次節に示す．

4.2 実験結果

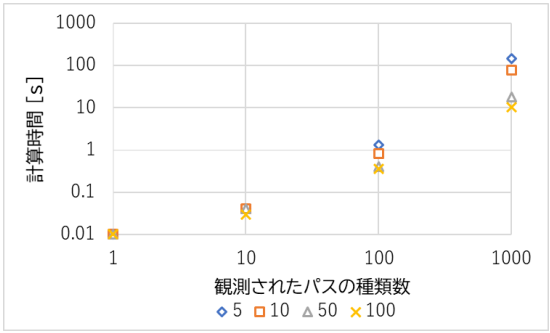
4.2.1 計算時間とメモリ使用量

クエリハプロタイプの長さを変化させた場合と，事前に観測されているアラインメントのパスの種類数を変化させた場合のそれぞれについて，ゲノムグラフの複雑さ（1 塩基多型の出現頻度）を変えながら計算時間とメモリ使用量を計測する実験を行った．計測は GNU 1.7 バージョンの time コマンドを利用した．すべての実験において，事前に観測されているアラインメントの塩基列長は 1000 塩基とし，コストパラメータは頂点やアラインメントによらず一定（置換・挿入・欠損・組み換えのコストすべて 1）とした．

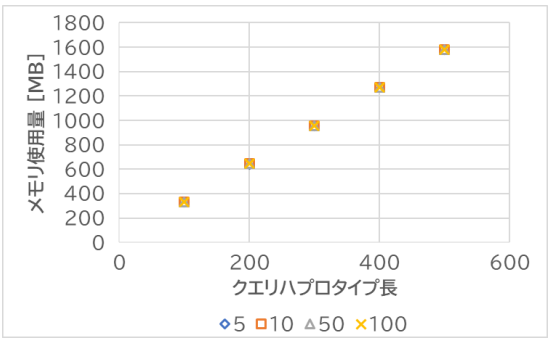
図 4.1 に実行時間とメモリ使用量を計測した結果を示した．図 4.1(a) からわかるように，実行時間はゲノムグラフの複雑さによらずクエリハプロタイプの長さに対して線形の時間がかかる．これは 3.6 節で示した通りである．また，図 4.1(b) は複雑なゲノムグラフであるほどパスの種類数の増加に対して計算時間が大きく影響されることを示している．これは，ゲノムグラフの分岐に線形比例する数の頂点上で組み換えを考えた計算を行わなければならないからである．なお，本実験においては 3.5.2 節で説明した高速化手法を用いている．コス



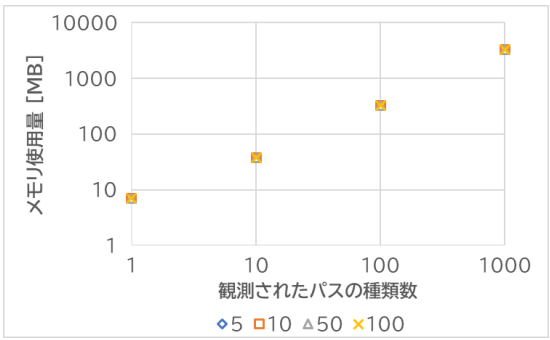
(a) 実行時間 1



(b) 実行時間 2



(c) メモリ使用量 1



(d) メモリ使用量 2

図 4.1　系列の項目に記載されている数字は，ゲノムグラフ内で分岐が起きる頻度を表している．例えば，青の菱形は 5 塩基に 1 箇所の頻度で 1 塩基多型が存在するゲノムグラフを表している．左：事前に観測されたアラインメントのパスの種類数を 100 と固定し，クエリハプロタイプの長さを 100 から 500 まで変化させた場合の実験結果を表している．右：クエリハプロタイプの長さを 100 と固定し，事前に観測されたアラインメントのパスの種類数を 1 から 1000 まで変化させた場合の実験結果を表している．こちらは両対数グラフとなっている．

トパラメータによっては，3.5.2 節の高速化手法を活用できない．その場合は本実験よりもはるかに長い実行時間がかかってしまうことが推定される．

図 4.1(c) と図 4.1(d) は，メモリ使用量はクエリハプロタイプの長さとパスの種類数にそれぞれ線形比例するだけの時間がかかることを示している．また，ゲノムグラフの複雑さには影響されないことも示している．これは 3.6 節で示した通りであるといえる．

4.2.2　アラインメント精度

従来手法の POA と比較してアラインメントの精度がどの程度異なるか実験を行った．POA は置換・挿入・欠損のコストをすべて 1 とし，提案手法は置換・挿入・欠損・組み換

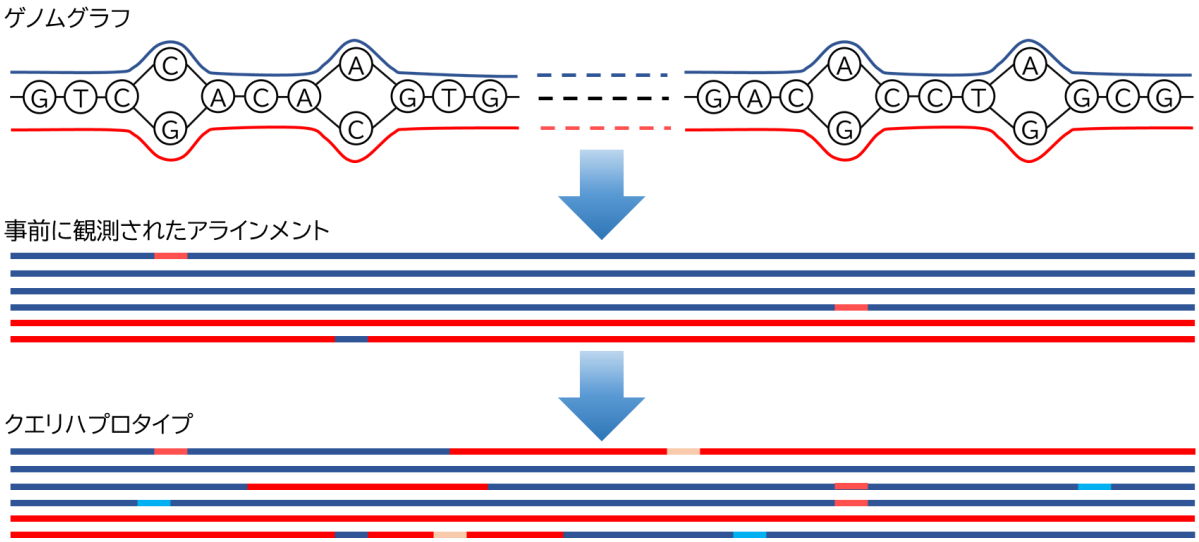


図 4.2　精度実験のために用いた実験データの生成フローを示した．最初にゲノムグラフを生成した．ゲノムグラフの上側を通るパス (青) をメジャーパス，下側を通るパス (赤) をマイナーパスとした．次にアラインメントを生成した．メジャーパスとマイナーパスのどちらかを選び，置換確率に基づいて分岐の進む方向を逆転させたものを複数生成した．青の直線に混じる小さな赤と赤の直線に混じる小さな青が置換を表している．最後にクエリハプロタイプを生成した．組み換え確率に基づいて複数のアラインメントを組み換えた後，エラー確率に基づいて置換を発生させた．例えば一番上のクエリハプロタイプは，前半がメジャー側で後半がマイナー側を通過しているため，組み換えが起こって発生したものだとわかる．また，色の薄くなっている部分が置換を表している．

えのコストをすべて 1 として計算を行った．

実験に用いたデータについて述べる．実験データの生成フローを図 4.2 に示した．実験に用いたデータは以下の条件を満たすように乱数を用いて生成した．実験に用いたゲノムグラフは全長が 100 塩基で 20 箇所に 2 分岐が存在する．事前に観測されたアラインメントは 100 本用意し，どのアラインメントも塩基長は 100 塩基とした．事前に観測されたアラインメントは大きく分けて 2 種類あり，1 つはすべての分岐においてメジャー側を通過するもので，もう 1 つはマイナー側を通過するものである．メジャー側を通過するアラインメントは全体の 7 割と設定した．各アラインメントは，置換確率に基づいて，各分岐の進む方向を変えた (メジャー方向とマイナー方向を逆転させた)．クエリハプロタイプは事前に観測されたアラインメントに対して，置換と組み換えを起こして生成した．クエリハプロタイプを生成する際の置換はエラー確率に，組み換えは組み換え確率に基づいて起こした．

置換確率が高ければ，事前に観測されたアラインメントのパスの種類数が多くなる可能性が高くなる．エラー確率が高いと，クエリハプロタイプは生成元となったアラインメントと異なる塩基列となっている可能性が高くなる．組み換え確率が高いと，クエリハプロタイプ

は生成元のアラインメントを複数持つ可能性が高くなる．

クエリハプロタイプの生成元となったアラインメントと全く同じアラインメントを計算によって得られることができた場合が実験における正解とし，頂点が 1 つでも異なっていたら間違いとした．本実験における計算精度 acc は以下の式で計算される．

$$acc = \frac{|\{q|q \in Q, q.orgpath = \mathsf{alignment}(G, q).path\}|}{|Q|} \tag{4.1}$$

クエリハプロタイプの集合を Q で表している．また，クエリハプロタイプ q の生成元となったパスを $q.orgpath$，q をゲノムグラフ G にアラインメントして得られたパスを $\mathsf{alignment}(G, q)$ と表している．なお，パス同士の等号が成り立つのは，パスに含まれる頂点数と頂点 ID の順が完全に一致している場合を意味する．一般的にアラインメントの精度は対応付けに成功した塩基の頻度から計算するが，そちらの指標を用いると POA と提案手法の差が見えにくいと判断し採用しなかった．POA は塩基単位でアラインメントを計算しているのに対し，提案手法では塩基列全体を通してのアラインメントを行っているため，今回はアラインメント全体が一致しているかどうかで評価することにした．

置換確率	POA	提案手法
0.0001	0.85	0.96
0.001	0.85	0.97
0.01	0.89	0.98
0.1	0.89	0.95

表 4.1　置換確率を変えた場合の計算精度

エラー確率	POA	提案手法
0.0001	1.0	0.98
0.001	1.0	0.97
0.01	0.89	0.98
0.1	0.18	0.67

表 4.2　エラー確率を変えた場合の計算精度

組み換え確率	POA	提案手法
0.0001	0.83	0.97
0.001	0.86	0.98
0.01	0.89	0.98
0.1	0.88	0.46

表 4.3　組み換え確率を変えた場合の計算精度

置換確率・エラー確率・組み換え確率をそれぞれ 0.0001 から 0.1 まで変化させて計測を行った結果を表 4.1 から表 4.3 に示した．表 4.1 に置換確率を変化させた場合の実験結果を，表 4.2 にエラー確率を変化させた場合の実験結果を，表 4.3 に組み換え確率を変化させた場合の実験結果を示した．変化させなかった側の確率はすべて 0.01 とした．

表 4.1 は，提案手法が事前に観測されているアラインメントの種類数が増加したとしても，高い精度でアラインメントを計算できることを示している．表 4.2 は，提案手法がリードにシーケンスエラーが入っていたとしても正しくアラインメントができることを示している．特にエラー確率が高い場合において，POA よりはるかに高い精度でアラインメントを計算することができている．表 4.3 は，提案手法が事前に観測されていない未知のパスに対してもアラインメントができること示している．ただし，組み換え確率が 0.1 の場合，提案手法が POA と比べて大きく精度が劣っている．これは組み換えによって，事前に観測されていないパスを通るアラインメントが多く生成されたためであると考えられる．本実験では，置換コストと組み換えコストを同じ値にしているので，組み換え確率とエラー確率が等しい時，提案手法が最もうまくアラインメントできると考えられる．提案手法において組み換えの確率が高い場合には組み換えのコストを小さくすることで，アラインメントの精度を改善できると考えられる．

実験に用いたクエリハプロタイプは，事前に観測されたアラインメントを元に生成した．POA はアラインメントを計算する際，事前に観測されたアラインメントを考慮しない手法である．一方で提案手法は事前に観測されたアラインメントを考慮する手法であるため，多くの条件で提案手法の方が高い精度でアラインメントを計算できたと考えられる．

なお，今回の実験では POA，提案手法共にすべてのコストパラメータを 1 として計算している．置換や組み換えに関する確率に基づいてコストパラメータを適切な値とすることにより，従来手法，提案手法共にアラインメントの精度を上げることが可能であると考えられる．

4.3 考察

4.3.1 関連研究との比較

関連研究と提案手法の関係性について述べる．表 4.4 に主な関連研究と提案手法との差異をまとめた．比較の詳細については以下で述べる．

POA との比較

POA は事前に観測されたアラインメントを考慮せずに計算を行う．つまり，POA は連鎖の情報を一切活用しないモデルであり，提案手法において組み換えのコストを 0 にした状況と同じである．実験において組み換えの確率が高い場合に提案手法より POA の方が精度が高かったのは，組み換えに対するコストの値が 1 よりも 0 とする方が，クエリハプロタイプの生成モデルに適合していたからであると考えられる．

手法名	用いているコスト	メリット・デメリット
POA	置換・挿入・欠損	高速に計算が可能 パスの頻度を考えていない
vg map	置換・挿入・欠損・パス	パスのコストを正確に計算している 最適解が求まる保証がない
提案手法	置換・挿入・欠損・組み換え	組み換えを考慮した最適解を求められる 計算量が従来手法と比べ大きい

表 4.4　関連研究との比較

Rosen らの確率モデルとの比較

提案手法の確率モデルにおいて，挿入と欠損の状態を取り除き，置換を許さないような制約を加えると，Rosen らが考えている確率モデルと等しくなる．提案手法のアルゴリズムでは，ゲノムグラフ上のパスを与えられたとき，そのパスを構築できる確率モデル上のすべてパターンの中で最も組み換え回数が少ないもののみを考えている．一方で，Rosen らのアルゴリズムではパスを構築できるパターン全てを考えている．Rosen らのアルゴリズムでは事前に観測されているアラインメントに置換・挿入・欠損が含まれていないという状況を考えているので，同じ部分パスを持つアラインメントをまとめて計算することができ，すべての組み換えのパターンについて考慮することが可能となっている．

vg との比較

vg のアラインメント計算では POA を行った後に，アラインメントのパスに対してコストを加算している．しかし，この手法ではリードが希少なパス上にマッピングされ，アラインメントスコアには高いパスのコストが加算されてしまう可能性がある．例えば vg map ではサブグラフに対してアラインメント候補は 1 つであるため，本来サブグラフの中で最良であるはずのアラインメントがアラインメントの候補に含まれないといった状況が起こりうる．提案手法ではアラインメント時に組み換えのコストを考慮しているので，vg では見つけることのできなかったアラインメントを発見できる可能性がある．

4.3.2　提案手法の限界

本研究では組み換えによるパスのコストはそのパスを得られる組み換えのパターンの中で最も組み換えの回数が少ないパターンのみを考えている．つまり，同じパスが複数存在していたとしても，そのパスの頻度の違いについては考慮できていない．とりうるパスすべてを事前に観測していた場合は，POA で計算した場合と全く同じアラインメント結果が得られ

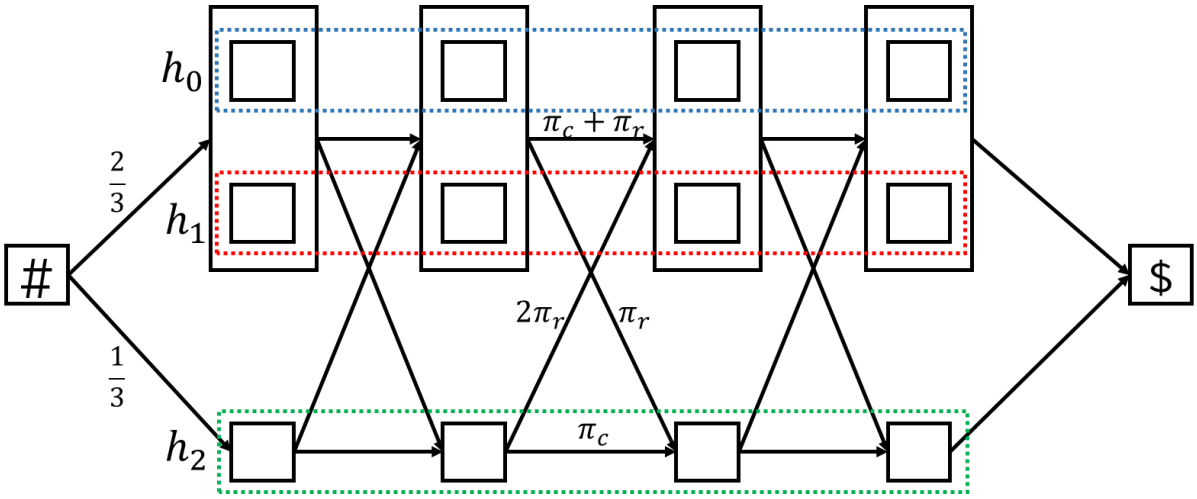


図 4.3　提案手法の確率モデルの中で，組み換えに関する辺と状態のみを示している．アラインメント h_0,h_1,h_2 が観測されており，h_0 と h_1 は同一のアラインメントを表している．h_0 と h_1 は確率モデルの中では 1 つのアラインメントとして統合されている．h_0 と h_1 を統合したことにより，遷移確率が変化している．例えば，h_0 と h_1 を統合したアラインメント内での遷移確率は，組み換えが起きない確率と，h_0,h_1 間で組み換えが起きる確率の和であるため $\pi_c + \pi_r$ となる．

る．つまり，パスの情報を考慮していない状況と同じになる．

多くの種類のパスが事前に観測されており，それぞれのパスの種類に関しては頻度の差があるという状況に関しては，本手法でうまく対処することができない．Rosen らのアルゴリズムでは，考えうるすべての組み換えのパターンを考慮しているため，事前に観測されたパスの頻度について考慮できているが，本手法ではあるパスについて事前に観測されているかどうかのみを考えているため，パスの頻度については考慮することができない．

パスの頻度情報を考慮したアラインメントを行うための 1 つの解決策としては，3.5.1 節で紹介したアラインメントの統合と同時に，確率モデルの遷移確率を変動させる手法が考えられる．

図 4.3 に，アラインメントを統合した場合の確率モデルの例を示した．統合したアラインメントの本数に応じて，組み換えに関する遷移確率を変動させることにより，パスの頻度についても考慮することが可能となる．

4.3.3　提案手法の応用

提案手法は，HLA 領域のような多様性が観測されている領域に対するアラインメント計算において従来手法よりも有効であると考えられる．例えば，POA を用いた場合，多様性が観測される領域では完全一致する部分文字列の種類数が非常に多くなるため，実際には観

測されていない希少なパス（組み換えが起きたことにより生成されたと解釈されるパス）に対するアラインメントのスコアが非常に良い値になる可能性がある．しかし，提案手法であれば，組み換えを行わなければ観測できないパスにはコストを加算することができるので，希少なパスに対するアラインメントに対して必要以上に良いスコアを与えることはない．

提案手法における各 ProfileHMM にアノテーションをつけることができれば，アラインメントを行うことで新しい知見が得られる可能性がある．例えば，地域ごとにヒトゲノムの ProfileHMM を構築しておけば，ハプロタイプのアラインメントを計算してどの ProfileHMM を通過しているかを確認することで，そのハプロタイプを持つヒトの先祖がどの地域に住んでいたかを高い精度で解析できるようになると考えられる．

第 5 章

総括

5.1　まとめ

本研究では，既に観測されているアラインメントの組み換えを考慮したゲノムグラフのアラインメント計算アルゴリズムを提案した．従来手法ではアラインメントの計算に事前に観測されているアラインメントの情報を活用しておらず，希少な変異の組み合わせやシーケンスエラーを含むリードに対しても高いアラインメントスコアを与えてしまう可能性があった．

提案手法を用いることで，組み換えを考慮したアラインメントを多項式時間内に計算でき，上記の問題を解決できることを示した．

5.2　今後の展望

今後は提案手法の実用性を示すため，実データを用いてアルゴリズムの検証を行おうと考えている．その際は vg のアラインメント用のコマンドに提案手法を組み込み，実験を行いたいと考えている．

また，3.2 節で示した理想のアラインメントと提案手法で計算できるアラインメントが実験的にどの程度異なるのかを計測したいと考えている．

謝辞

本研究を行うにあたり，さまざまなご指導を頂きました清水佳奈教授に深く感謝いたします．

研究に関して何度も相談をしてくださった東京大学笠原研究室の鈴木創様に心から感謝いたします．

関連研究の調査を共に行ったゲノムグラフ研究会の皆様，どうもありがとうございました．

最後に，清水研究室の皆様，研究だけでなく様々な面でお世話になりました．ありがとうございました．

参考文献

[1] Tools for working with genome variation graphs. https://github.com/vgteam/vg, 2018. URL accessed December 26, 2018.

[2] Richard Durbin. Efficient haplotype matching and storage using the positional burrows–wheeler transform (pbwt). *Bioinformatics*, Vol. 30, No. 9, pp. 1266–1272, 2014.

[3] Sean R. Eddy. Profile hidden markov models. *Bioinformatics (Oxford, England)*, Vol. 14, No. 9, pp. 755–763, 1998.

[4] Erik Garrison, Jouni Sirén, Adam M Novak, Glenn Hickey, Jordan M Eizenga, Eric T Dawson, William Jones, Michael F Lin, Benedict Paten, and Richard Durbin. Sequence variation aware genome references and read mapping with the variation graph toolkit. *bioRxiv*, 2017.

[5] Christopher Lee, Catherine Grasso, and Mark F Sharlow. Multiple sequence alignment using partial order graphs. *Bioinformatics*, Vol. 18, No. 3, pp. 452–464, 2002.

[6] Na Li and Matthew Stephens. Modeling linkage disequilibrium and identifying recombination hotspots using single-nucleotide polymorphism data. *Genetics*, Vol. 165, No. 4, pp. 2213–2233, 2003.

[7] Mikko Rautiainen and Tobias Marschall. Aligning sequences to general graphs in o(v + me) time. *bioRxiv*, 2017.

[8] Yohei Rosen, Jordan Eizenga, and Benedict Paten. Modelling haplotypes with respect to reference cohort variation graphs. *Bioinformatics*, Vol. 33, No. 14, pp. i118–i123, 2017.

[9] Anne-Kathrin Schultz, Ming Zhang, Ingo Bulla, Thomas Leitner, Bette Korber, Burkhard Morgenstern, and Mario Stanke. jphmm: Improving the reliability of recombination prediction in hiv-1. *Nucleic acids research*, Vol. 37, No. suppl_2, pp. W647–W651, 2009.

[10] Jouni Sirén. Indexing variation graphs. *CoRR*, Vol. abs/1604.06605, , 2016.

[11] Jouni Sirén, Erik Garrison, Adam M. Novak, Benedict Paten, and Richard Durbin.

Haplotype-aware graph indexes. *CoRR*, Vol. abs/1805.03834, , 2018.

[12] Esko Ukkonen. Algorithms for approximate string matching. *Information and control*, Vol. 64, No. 1-3, pp. 100–118, 1985.